Project Report: Credit Card Fraud Detection

1. Introduction and Goal

- **Introduction**:
  - The rapid growth of digital transactions has led to an increase in credit card fraud, posing significant financial risks to consumers and institutions. Effective fraud detection systems are essential to mitigate these risks.
  - This project leverages machine learning techniques to develop and evaluate models for detecting fraudulent transactions in a dataset of credit card transactions.
- **Goal**:
  - The primary goal of this project is to build a robust credit card fraud detection system that can accurately identify fraudulent transactions.
  - To achieve this, multiple machine learning models will be trained and evaluated, with the best-performing model selected for deployment.

---

2. Dataset Overview

- **Dataset**:
  - The dataset contains credit card transactions, including 284,807 rows and 31 columns, where each row represents a transaction and columns represent various attributes of the transaction, including a 'Class' column indicating fraud (1) or non-fraud (0).

---

3. Data Exploration

- **Top and Bottom Rows**:
  - Displayed the first and last 5 rows to understand the data structure and contents.
- **Data Shape**:

    - The dataset comprises 284,807 transactions (rows) and 31 attributes (columns).
- **Data Information**:
    - All columns are numeric, with no missing values.

---

4. Data Preprocessing

- **Null Values**:
    - Confirmed there are no null values in the dataset.
- **Feature Scaling**:
    - Applied standard scaling to the 'Amount' column to normalize transaction amounts.
- **Duplicate Values**:
    - Identified and removed 1,081 duplicate rows, resulting in 283,726 unique transactions.
- **Time Column**:
    - Converted the 'Time' column from float to integer type for consistency.

---

5. Handling Imbalanced Data

- **Class Distribution**:
    - Identified significant class imbalance with 283,253 non-fraudulent transactions and only 473 fraudulent transactions.
- **Undersampling**:
    - Balanced the dataset by undersampling non-fraudulent transactions to match the number of fraudulent transactions, resulting in 946 transactions (473 fraud and 473 non-fraud).

---

6. Feature Matrix and Target Variable

- **Feature Matrix (X)**:
    - Extracted all columns except 'Class'.
- **Target Variable (Y)**:
    - Used the 'Class' column as the target variable.

7. Train-Test Split

- Split the balanced dataset into training (80%) and testing (20%) sets to ensure the model is evaluated on unseen data.

---

8. Model Training and Evaluation

- **Logistic Regression (LR)**:
  - Achieved training accuracy: 94.18%
  - Test accuracy: 93.68%
  - Precision, Recall, and F1 scores for both training and test sets indicate good performance.
- **Decision Tree (DT)**:
  - Training accuracy: 94.18%
  - Test accuracy: 88.42%
  - Slightly lower performance compared to logistic regression.
- **Random Forest (RF)**:
  - Training accuracy: 100%
  - Test accuracy: 92.63%
  - Excellent performance on training data but slightly lower on test data due to potential overfitting.

---

9. Model Comparison

- Compared the accuracy of the three models using a bar plot, highlighting logistic regression as the best-performing model for this dataset.

---

10. Model Deployment

- **Saving the Model**:
  - Saved the logistic regression model using Joblib for future predictions.
- **Loading and Prediction**:

- Loaded the saved model and demonstrated a prediction, indicating whether a transaction is legitimate or fraudulent.

---

11. Conclusion

- Successfully implemented and evaluated three machine learning models for credit card fraud detection.
- Logistic regression model showed the best performance and was selected for deployment.
- Future improvements could include exploring more advanced techniques for handling imbalanced data and incorporating additional features or data sources.

---

12. Future Work

- **Explore Advanced Techniques**:
  - Implement more sophisticated methods like SMOTE (Synthetic Minority Over-sampling Technique) for balancing the dataset.
- **Feature Engineering**:
  - Investigate new features that could improve model performance.
- **Model Optimization**:
  - Fine-tune hyperparameters and explore other algorithms such as Gradient Boosting or Neural Networks.
